

# CDF/D0/CMS Database Experience

Lee Lueking  
Fermilab

BTeV Software Week  
June 23, 2004

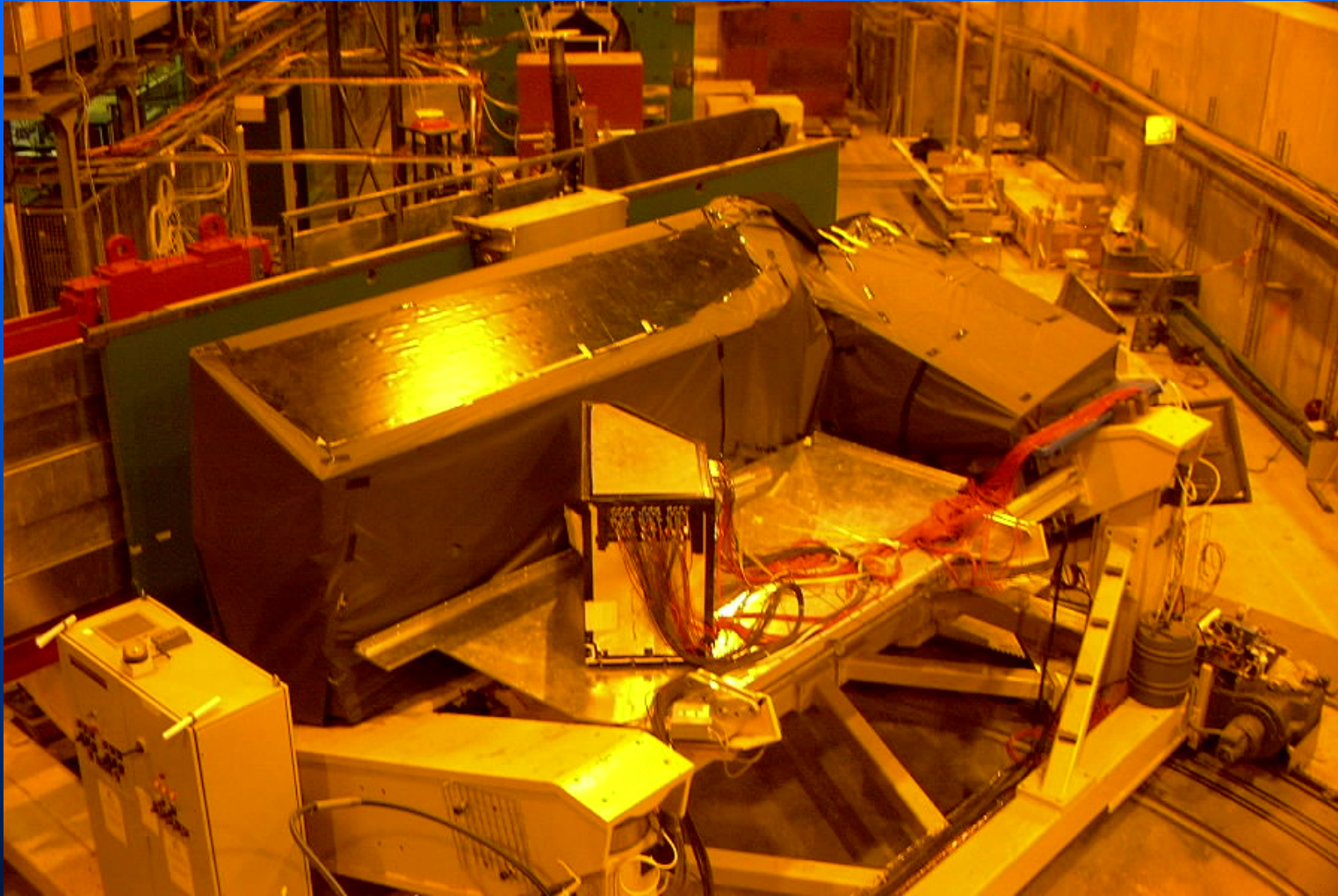
# Outline

- CMS:
  - HCAL Detector Database for the test beam
  - More General CMS DB Ideas
- Distributed DB access in Run II
  - D0 database server
  - CDF replication
  - The FroNtier Project
- Concerns and Conclusion

# USCMS DetDB Mission

- Collect use cases, establish the requirements, design and build a functioning database system for use in the 2004 HCAL test beam operation.
- Extend the experience gained in the HCAL project to additional Fermilab detector interests including EMU and PIXEL.
- Use the Test beam experience as a prototype for a full scale CMS detector database project for HCAL, EMU, and PIXEL, needed for full scale testing in 2006, and operation in 2007.
- And in the process...Establish close relationships with the CMS and LCG database teams, and IT at CERN, to understand the broader database landscape being established for the LHC. Become involved in the planning, and carefully define the role Fermilab can play in this area.

# CMS HCAL Testbeam @ CERN



June 23, 2004

BTeV SWS - Lee Lueking

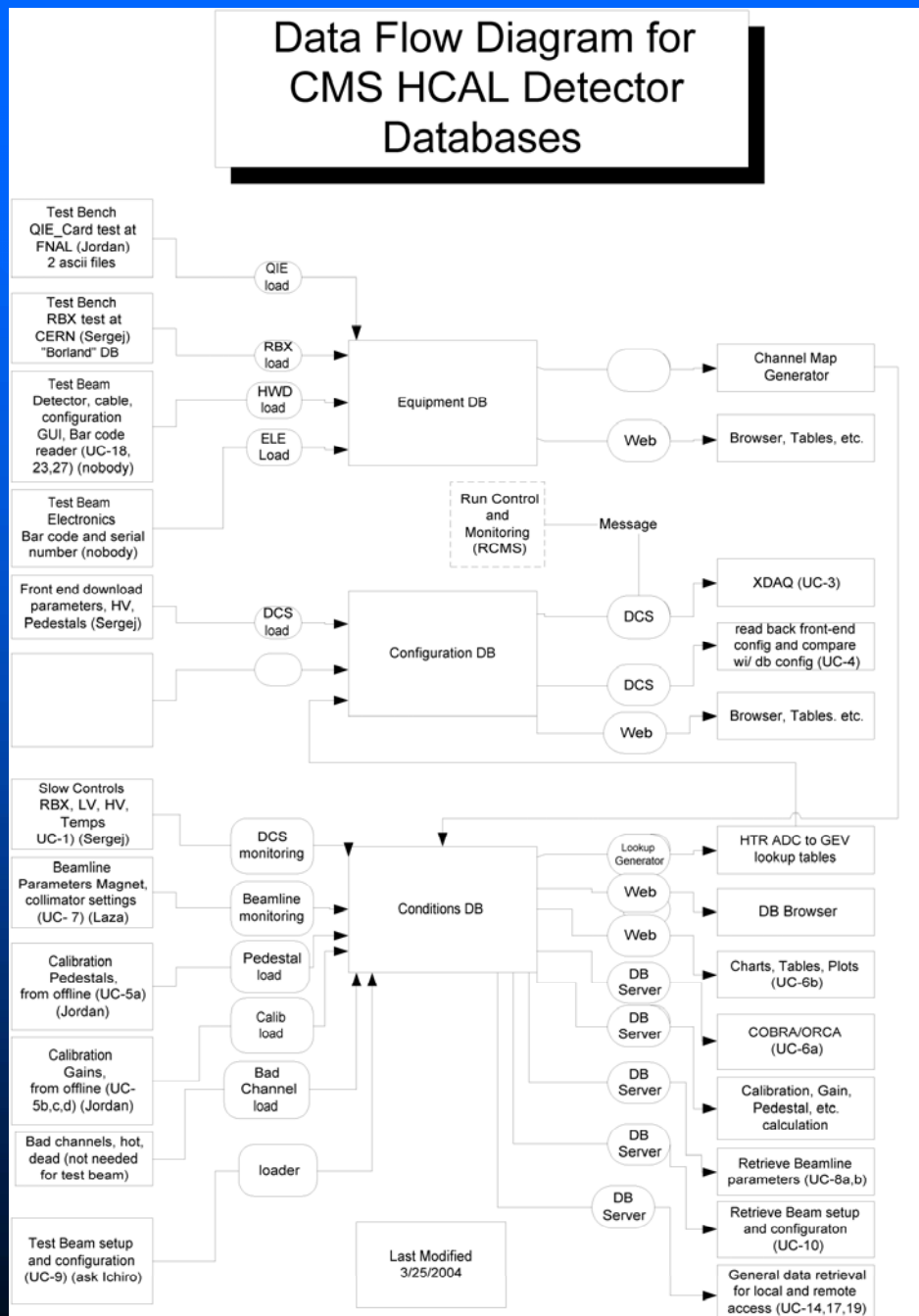
4

# Categories of DB Information

(Glossary)

- Construction
  - Test results for each detector component
  - Details of detector construction
- Equipment
  - Inventory of all detector components, and their locations
  - Details of channel mapping and electronics modules
- Configuration
  - Download constants for front-end electronics
  - Includes HV, LV, pedestals, etc.
- Conditions
  - Measured values for HV, LV, temps
  - Beam positions
  - Offline pedestal and gains
  - Generally, any value, changing with time, needed to understand the data.



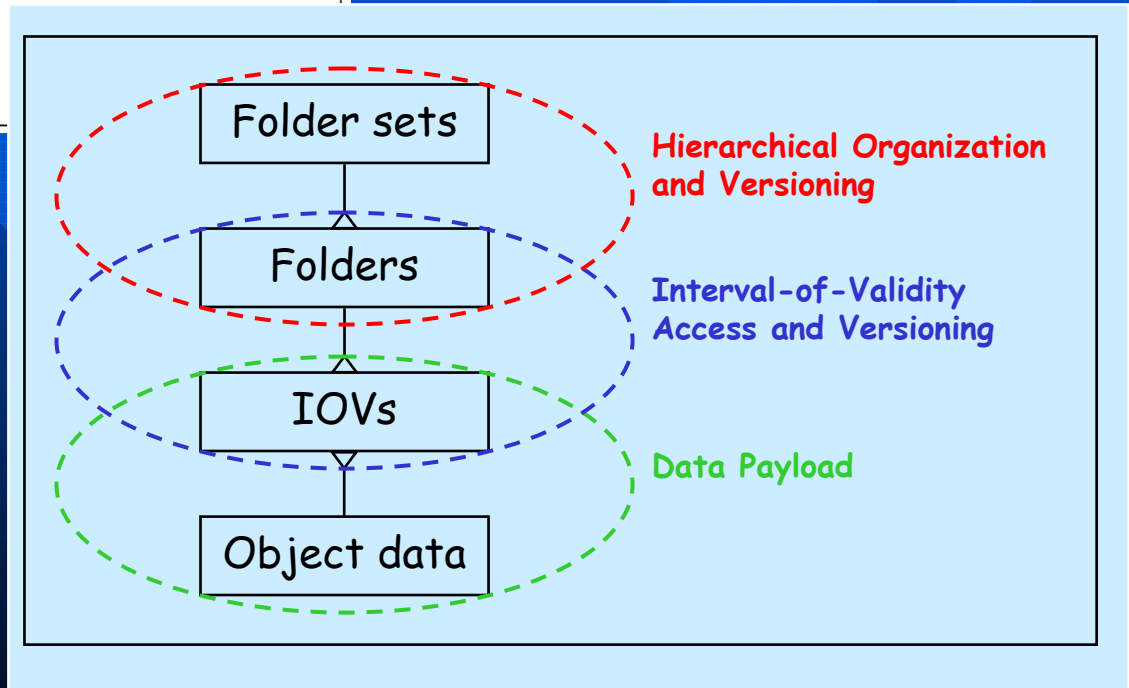
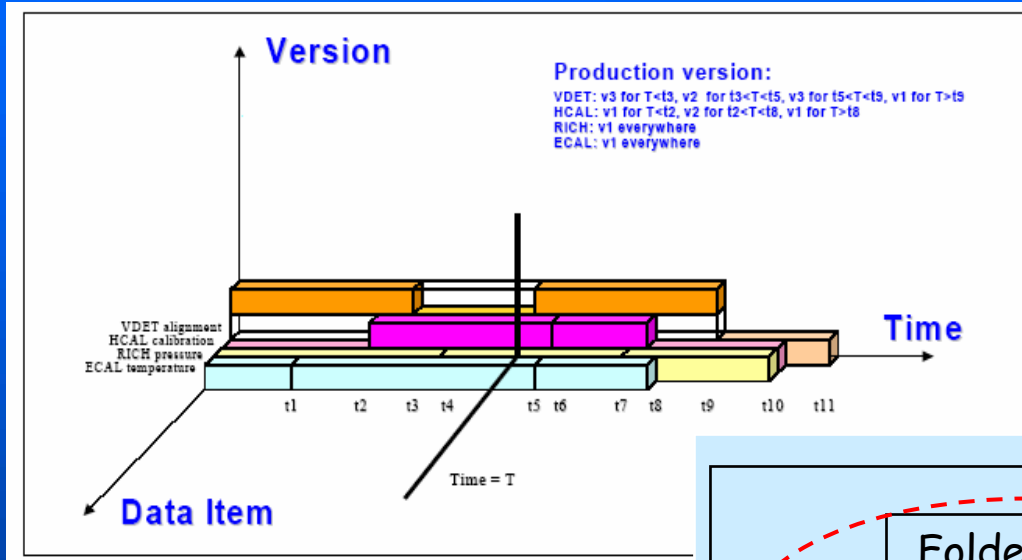


# Applications for HCAL Test Beam

The goal is to have a  
functioning Database  
by the end of June, 2004

This and additional  
details available at:  
<http://lynx.fnal.gov/uscms-db-wiki>

# LCG Conditions DB Concepts



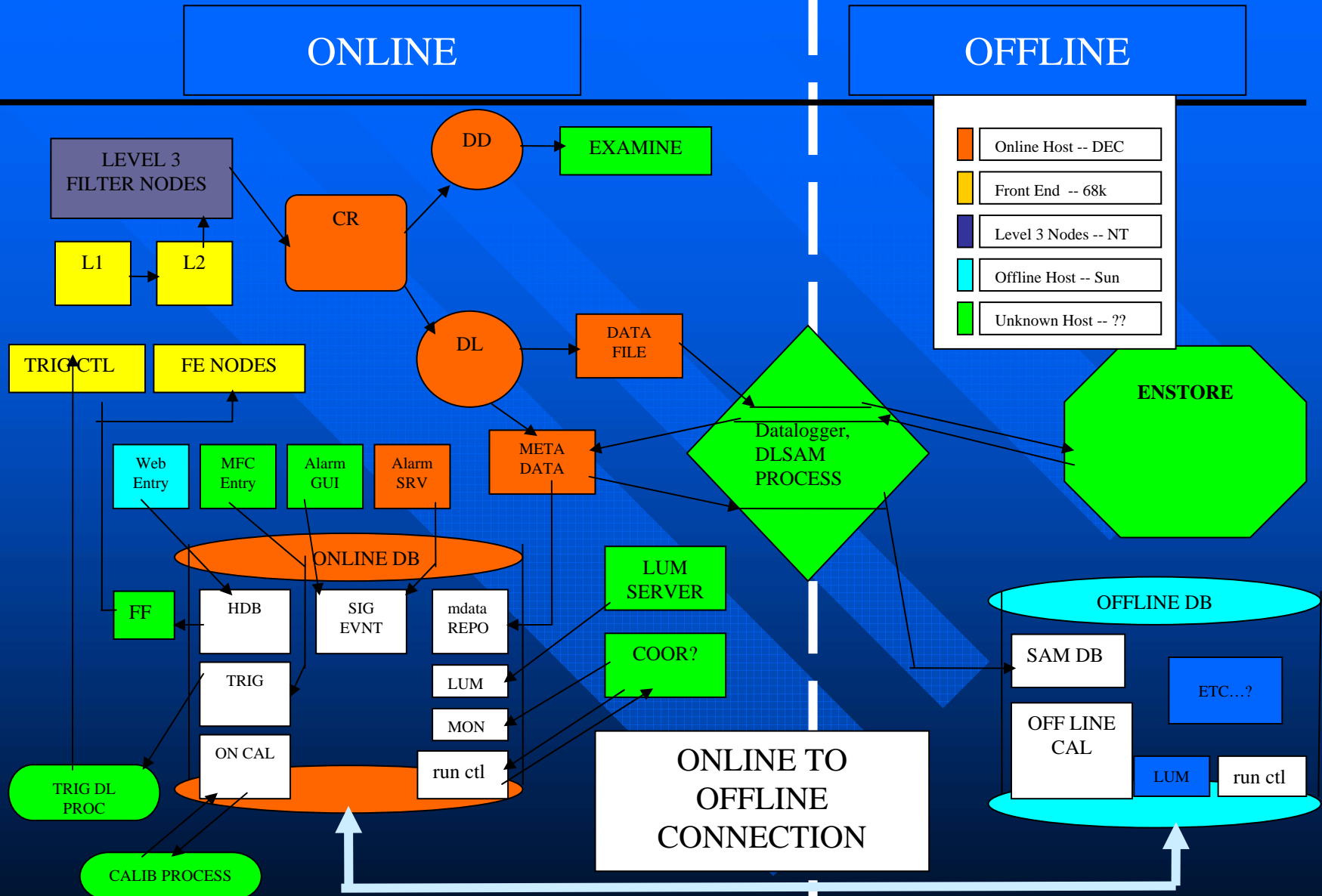
# Run II Experience Providing Scalable Database Access

For additional comparison of D0 and CDF database approaches please see Jack Cranshaw's talk presented at the LCG Conditions DB Workshop December 8, 2004

<http://agenda.cern.ch/fullAgenda.php?ida=a036470>



# DO Online to Offline Database Copy

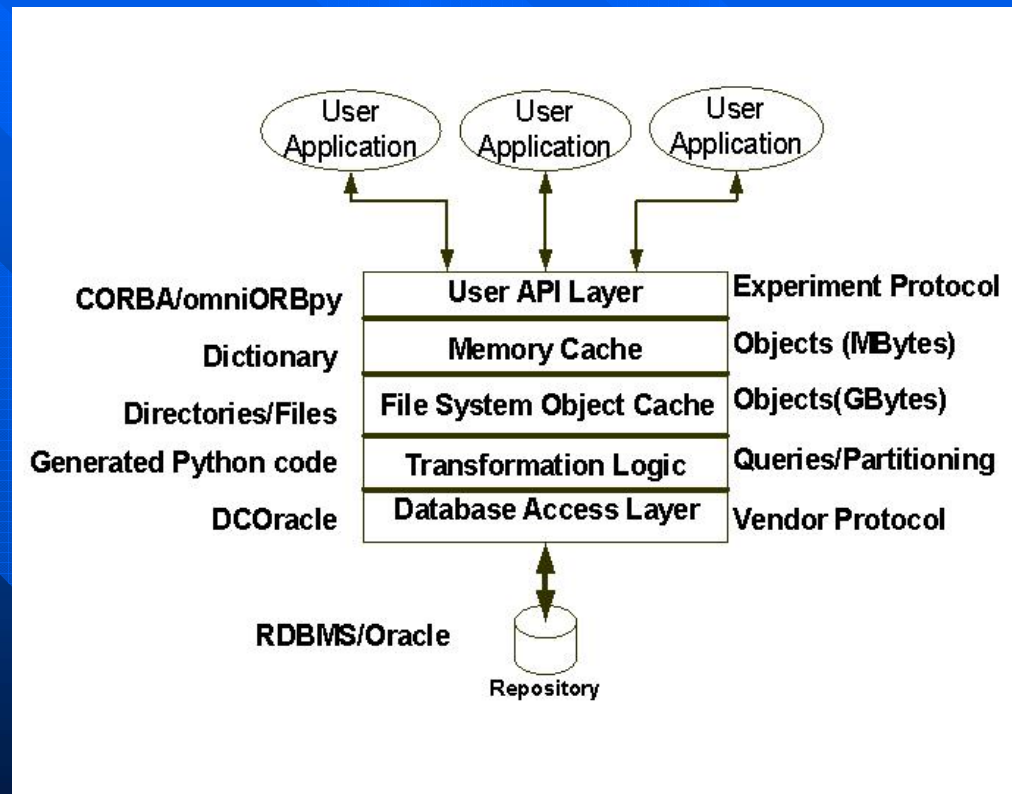


# DØ Offline Caching Server: DAN

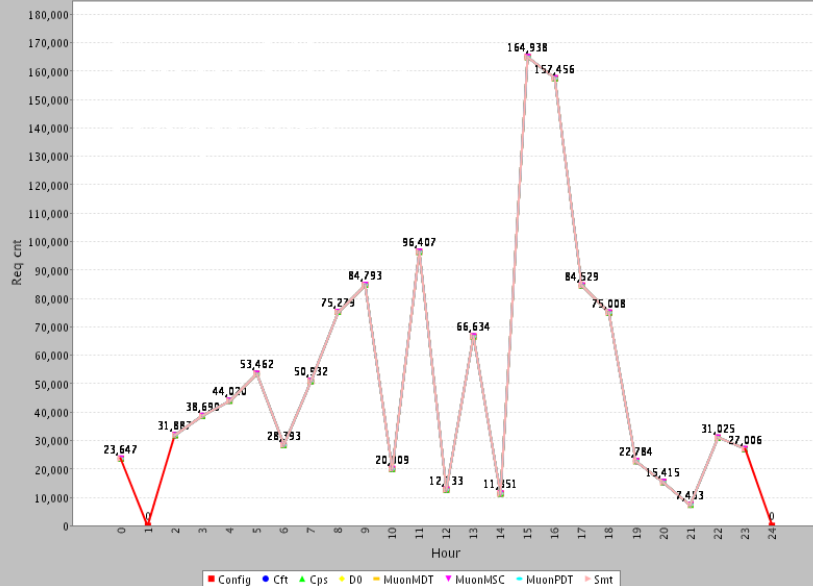
(Database Access Network)

- CORBA interface to Client apps
- Memory (L1) and Disk (L2) caching
- Connection management to Database
- Server has common code base with SAM DB server

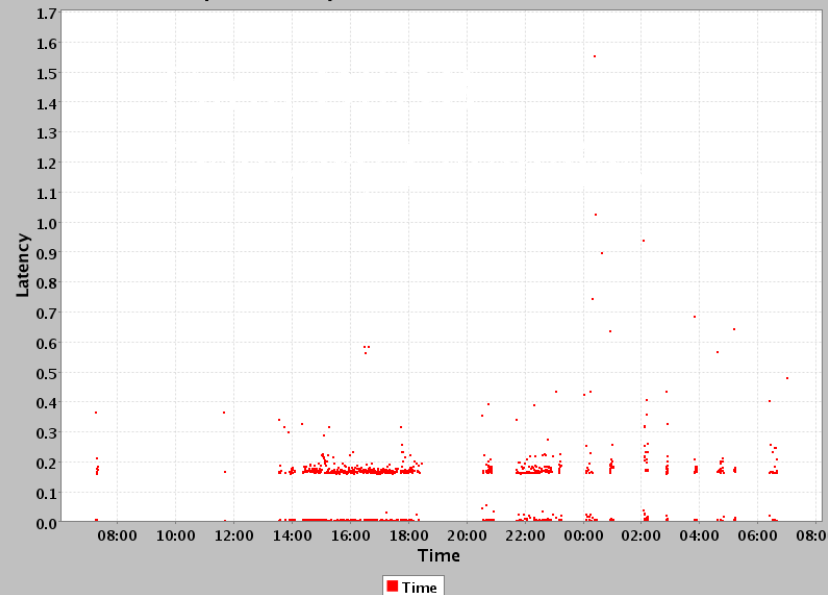
## Read-only DB access



CalibFarm N Req per srv for 24 hours for 2003-09-08 from 15:00

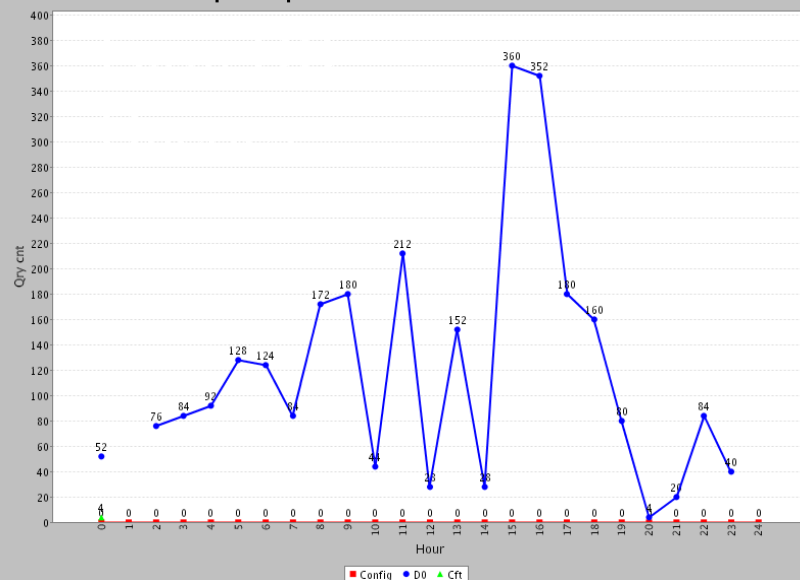


D0DbServer.farm\_prd - Latency Duration for 24 hrs for 2003-09-10 from 7:00



Database

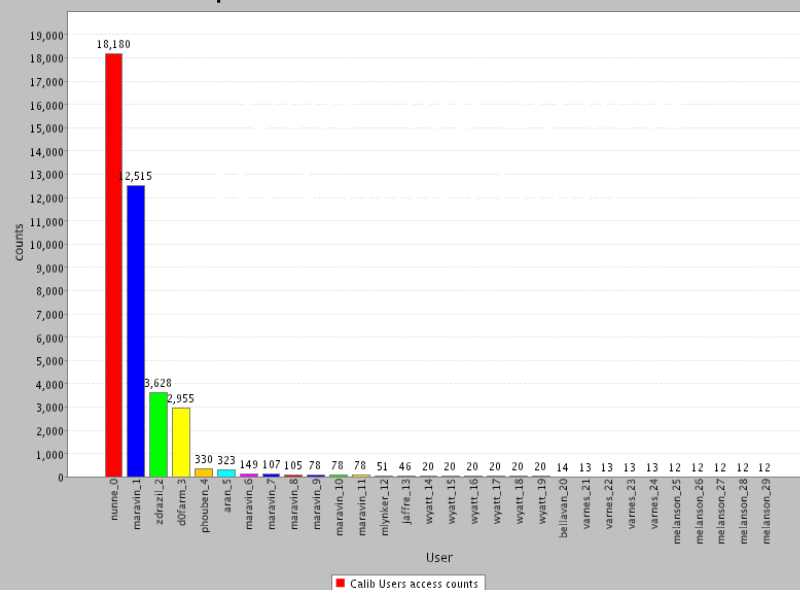
CalibFarm N queries per srv for 24 hrs for 2003-09-08 from 15:00



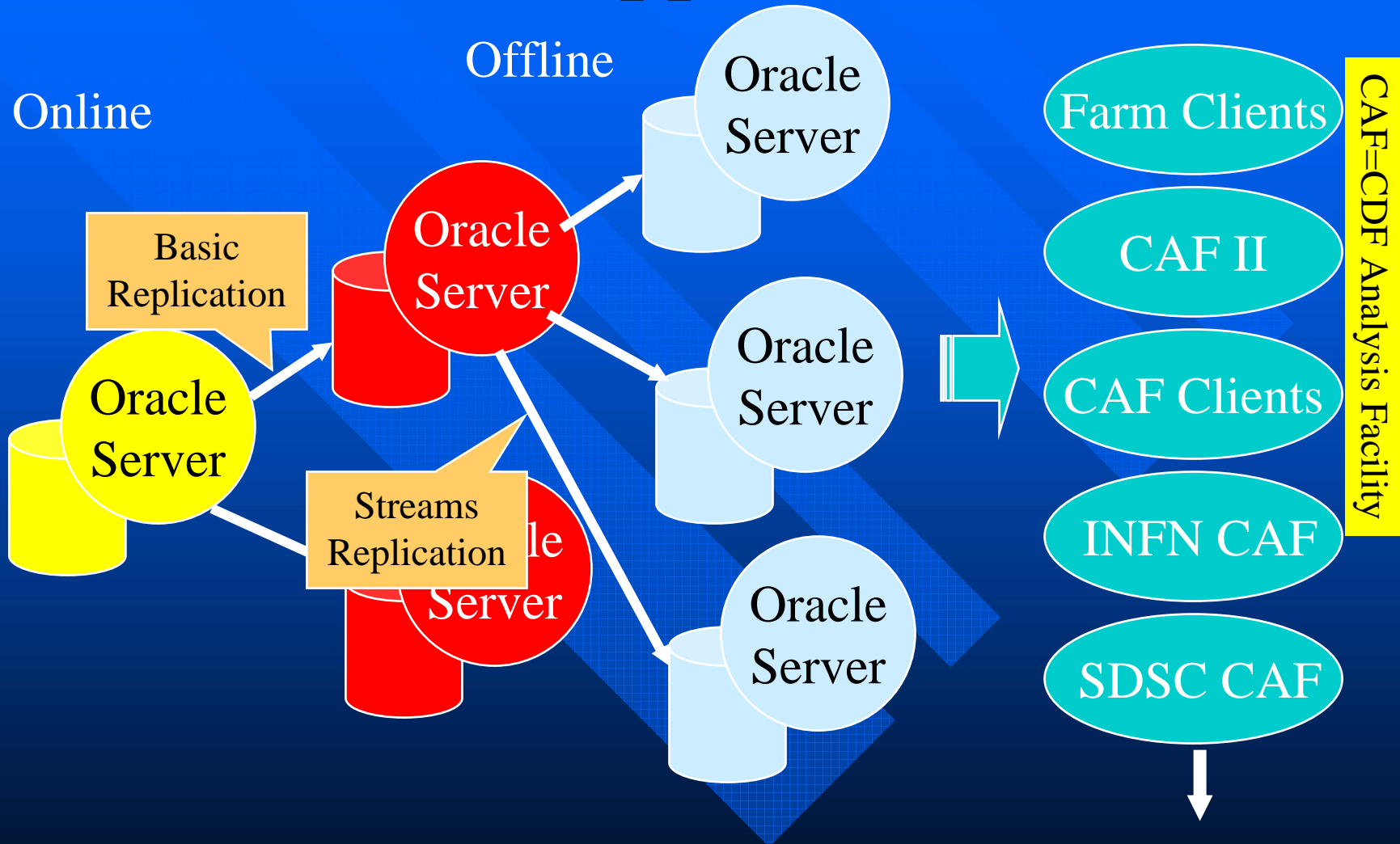
Servers

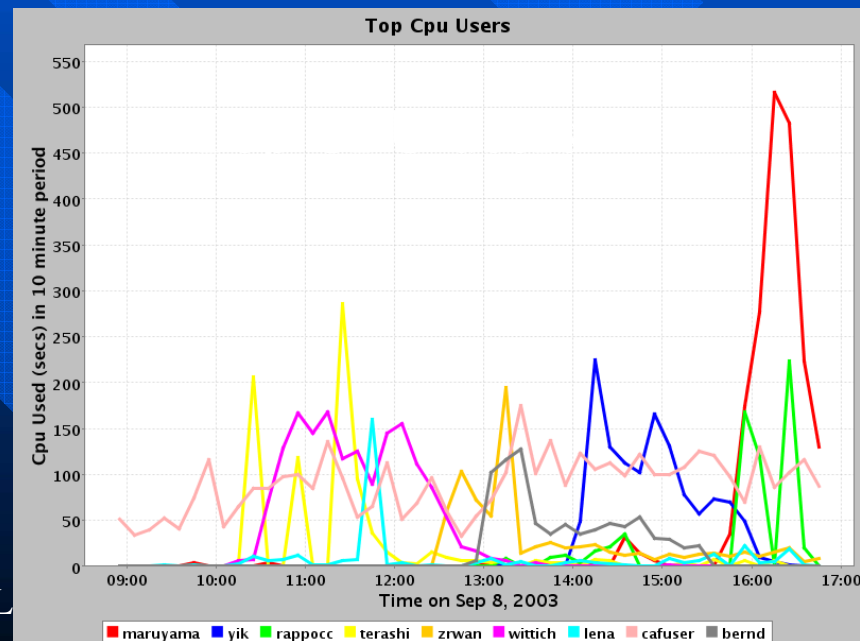
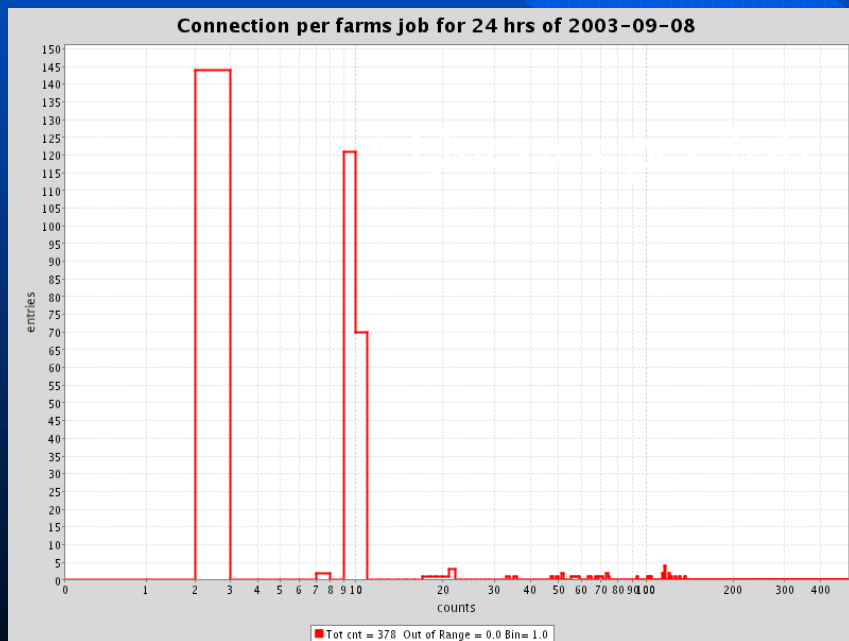
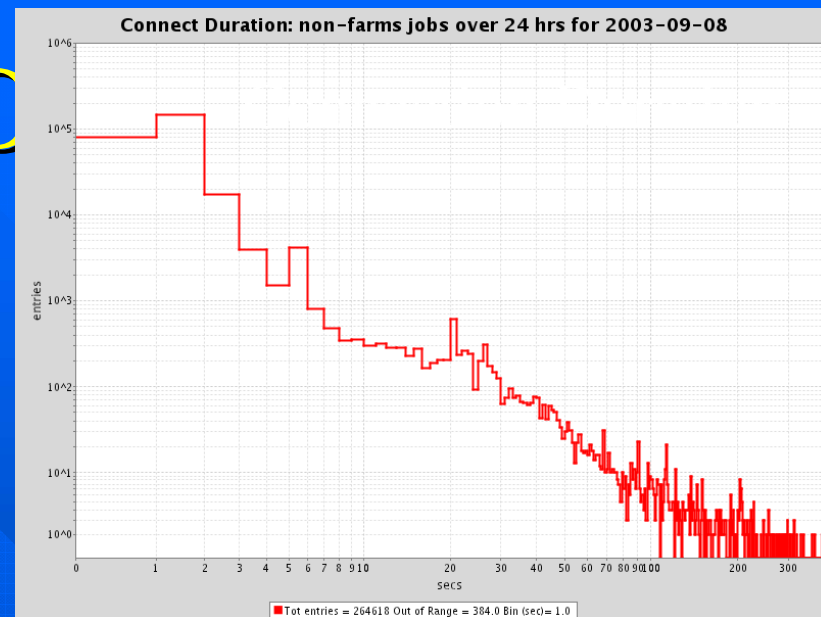
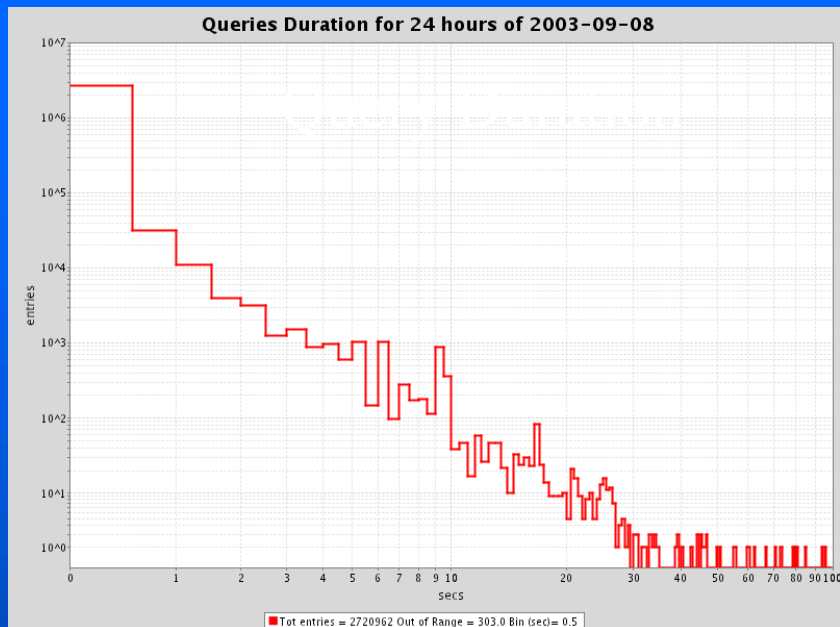
Fail over

CalibUser top 20 users for 24 hours for 2003-09-05 from 19:00



# CDF Approach





# FroNtier

**CDF: Barry Blumenfeld (JHU) , Dmitri Litvintsev, Petar Maksimovic (L)(JHU) , Mark Mathis(JHU), CD/APS: Sergey Kosyakov, Jim Kowalkowski, Lee Lueking (L), Marc Paterno, Steve White.**

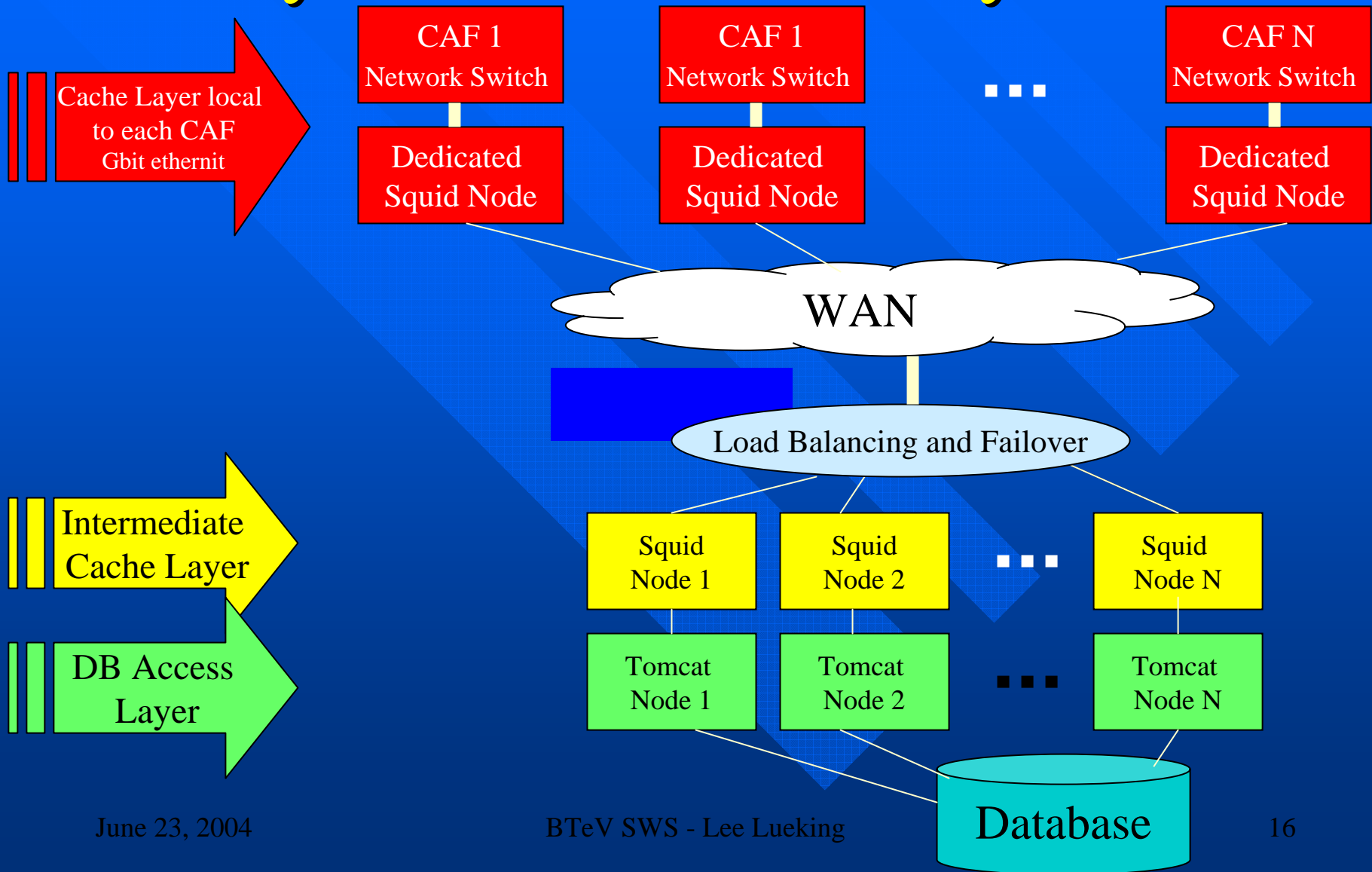
- **Goal: Assemble a toolkit, using standard web technologies, to provide high performance, scalable, database access through a multi-tier architecture.**
- **Pilot project Ntier tested the technology:**
  - Tomcat, HTTP, Squid
  - Client monitoring w/ existing CDF tools (udp messages)
- **Completed a working system for CDF vertical slice test with all needed components March 1. Additional features were added for production-like testing during May. Production version being built now.**
- **<http://whcdf03.fnal.gov/ntier-wiki/FrontPage>**



# A Multi-tier Architecture

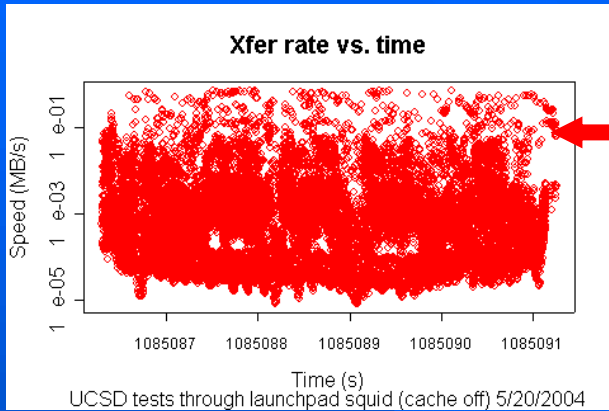
- Database (Oracle, MySQL, etc.)
- Database Access Layer
  - Tomcat: servlet management engine
  - JDBC: Database connection
  - Database Connection pool management
- Caching and proxy layer
  - Can provide some caching in the Tomcat server if needed
  - Squid: well known, widely used, highly configurable, caching proxy server
  - Others available, e.g. Netscape.
- Client:
  - Use HTTP library, like cURL / lib
  - Parse the payload, either base64, ascii, or binary format.

# System Hardware Layout



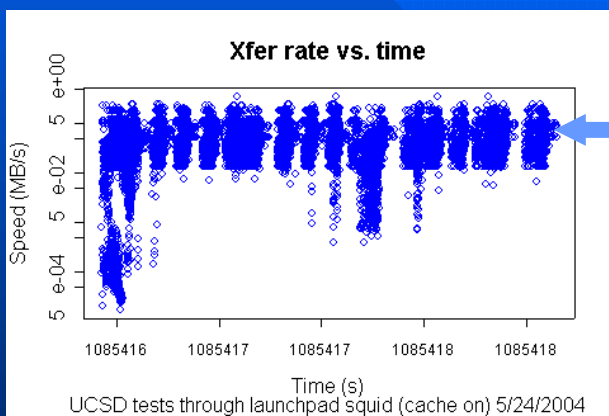
# FroNtier Testing at FNAL/SDSC

(San Diego Super Computing Center)



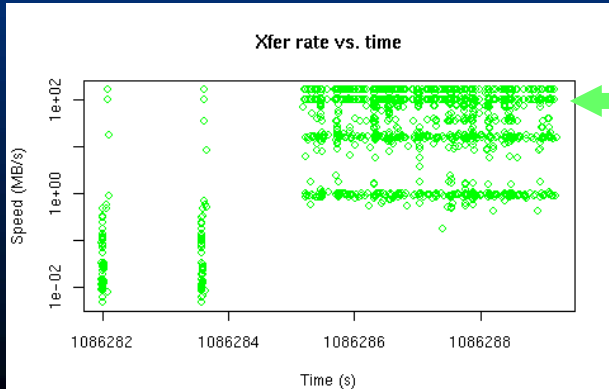
0.1 MB/s

No Caching



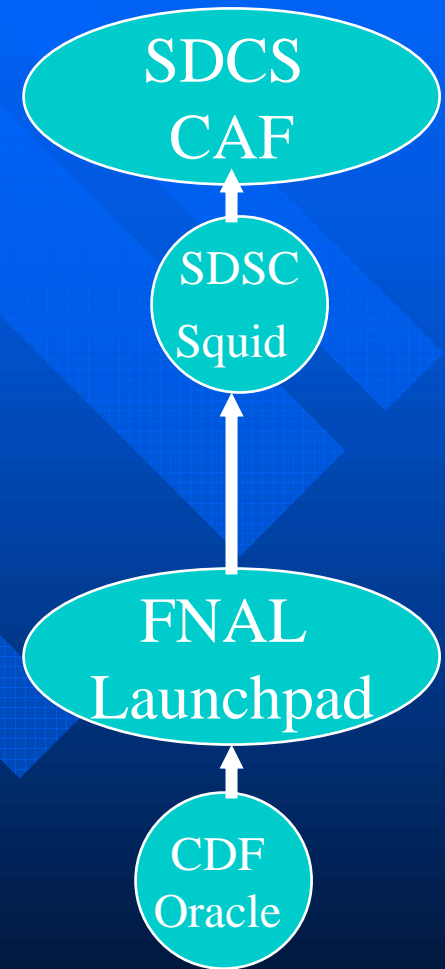
5 MB/s

Caching at FNAL  
Launchpad



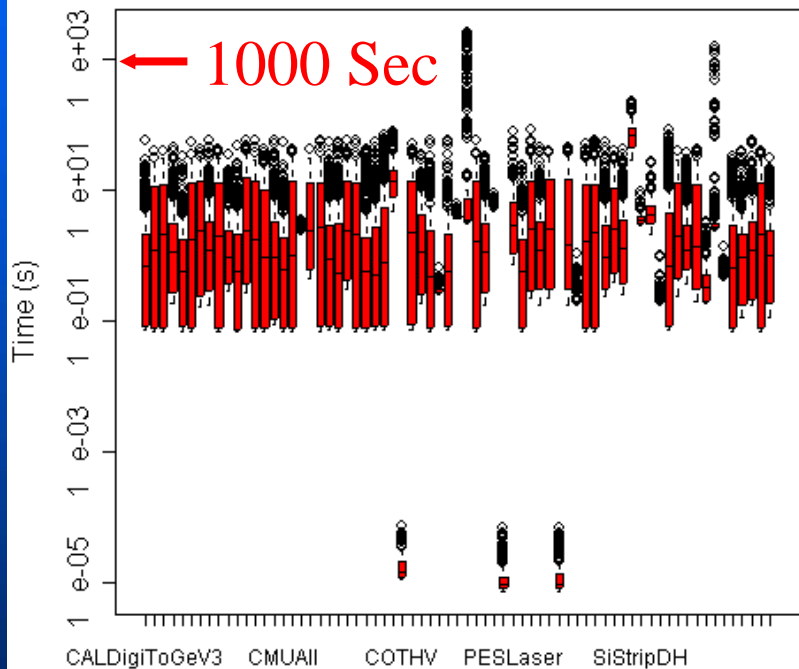
100 MB/s

Caching local to  
SDSC



# Client Side: FNAL/SDSC

Response time vs. Table Name

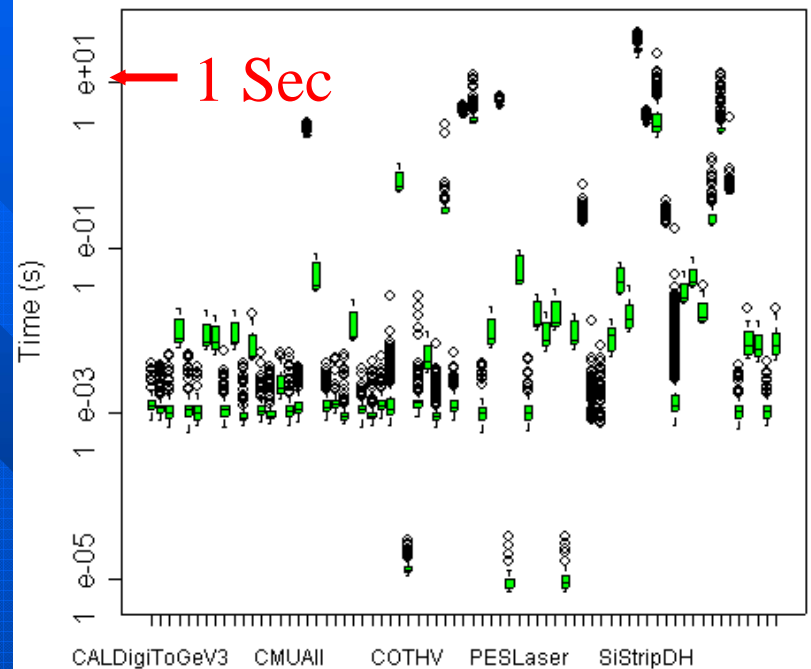


UCSD test

5/20/2004

No Caching

Response time vs. Table Name



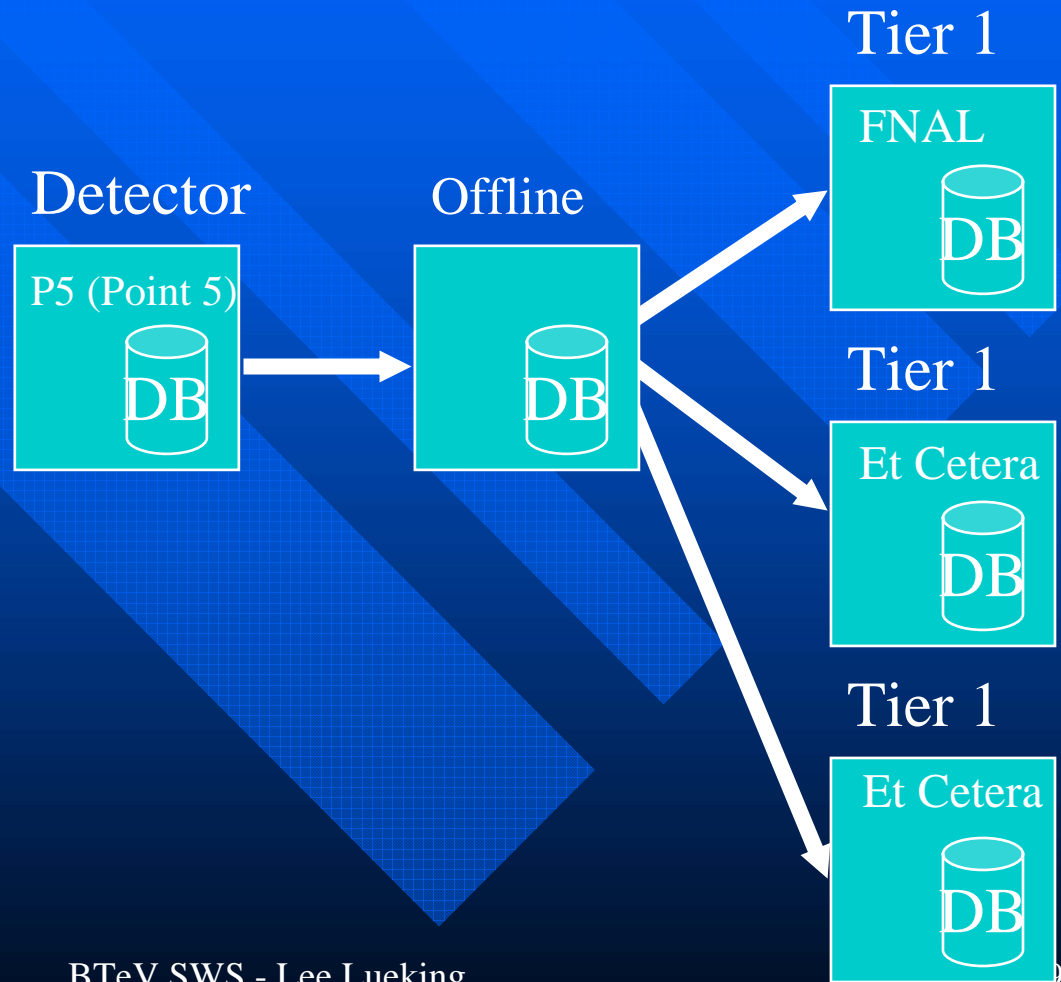
UCSD tests

id 6/07/2004

Caching at  
SDSC Squid

# Run II Database Distribution → CMS

- Conditions & other DB info needed for offline copied to Tier 0.
- DB info needed for offline analysis replicated to Tier 1 (?).
- Lightweight caching scheme (like FroNtier) used for access by Tier 1 and Tier 2 sites.



# Conclusion

- Work is ongoing to provide an Oracle-based DB system for the HCAL and EMU test beam operation at CERN this summer, and into the fall.
- The LCG Database project is trying to find common areas among all LHC experiments where solutions can be shared.
- The Run II experience with database delivery has shown that performance and scalability are achievable through replication and a multi-tier architecture.
- The FroNtier project uses “commodity” web tools to provide an extremely light-weight, multi-tier infrastructure for read-only data access.